

We claim:

1. A method for emulating network traffic, comprising:
 - initializing a finite state machine to a state keyword from at least two state keywords;
 - retrieving a value from the network frame;
 - determining if the value triggers a state keyword change; and
 - if the value triggers a state keyword change, changing the state keyword to another of the at least two state keywords.
2. The method of claim 1 further comprising:
 - determining if the value triggers an action routine; and
 - if the value triggers an action routine, performing an action.
3. The method according to any of claims 1 and 2 wherein the network frame has at least one protocol data unit having at least one field containing the value and the step of retrieving a value from the network frame further includes;
 - retrieving protocol knowledge of the data structure of the at least one protocol data unit enabling the extraction of the at least one field; and
 - extracting a value from the at least one field of the at least one protocol data unit.

4. The method according to any of claims 2 and 3 wherein the step of performing an action includes:

receiving at least one protocol data unit name and at least one value corresponding to at least one keyword associated with the at least one protocol data unit name;

retrieving protocol knowledge of the data structure of a network frame enabling the building of the at least one keyword into a network frame;

associating the at least one value with the at least one keyword;

placing the at least one protocol value into a memory device in a network frame data structure; and

transmitting the network frame to the network.

5. A computer system including an emulator for emulating network traffic, the computer system comprising:

a user interface;

a network connection;

an emulator system disposed between the user interface and the network connection, the emulator system including:

a protocol emulator connected to the user interface, including at least two state keywords defining each state of the protocol that the protocol may enter, the at least two state keywords representing a number of behavior states that a particular one of the protocol states may express, the particular one of the protocol states being associated

with a particular one of the at least two state keywords and a set of event keywords defining each state keyword representative of inputs that the protocol state may received while being associated with the particular one of the at least two state keywords, the protocol state operable to change its association to another one of the at least two state keywords in response to the inputs receive or actions encountered; and

a protocol decoder, connected to the protocol emulator and network connection, having means for receiving a network frame from the network connection and signaling a network event to the protocol emulator.

6. The computer system of claim 5 wherein the emulator system further includes:

a protocol encoder, connected to the protocol emulator for encoding network frame data received from the protocol emulator for transmission to the network.

7. A computer system for emulating network traffic, by a method comprising:

means for initializing a finite state machine to a state keyword from at least two state keywords;

means for retrieving a value from the network frame;

means for determining if the value triggers a state keyword change; and

if the value triggers a state keyword change, means for changing the state keyword to another of the at least two state keywords.

8. The method of claim 7 further comprising:

means for determining if the field value triggers an action routine; and

if the field triggers an action routine, means for performing an action.

9. The method according to any of claims 7 and 8 wherein the network frame has at least one protocol data unit having at least one field containing the value and the step of retrieving a value from the network frame further includes;

means for retrieving protocol knowledge of the data structure of the at least one protocol data unit enabling the extraction of the at least one field; and

means for extracting a value from the at least one field of the at least one protocol data unit.

10. The method according to any of claims 8 and 9 wherein the step of performing an action includes:

means for receiving at least one protocol data unit name and at least one value corresponding to at least one keyword associated with the at least one protocol data unit name;

means for retrieving protocol knowledge of the data structure of a network frame enabling the building of the at least one keyword into a network frame;

means for associating the at least one value with the at least one keyword;

means for placing the at least one protocol value into a memory device in a network frame data structure; and

means for transmitting the network frame to the network.

11. The computer system of claim 7 operable as a network analyzer.

12. A computer-readable medium whose contents cause a computer system to emulate network traffic, by performing the steps of:

initializing a finite state machine to a state keyword from at least two state keywords;

retrieving a value from the network frame;

determining if the value triggers a state keyword change; and

if the value triggers a state keyword change, changing the state keyword to another of the at least two state keywords.

13. The method of claim 12 further comprising:

determining if the field value triggers an action routine; and

if the field triggers an action routine, performing an action.

14. The method according to any of claims 12 and 13 wherein the network frame has at least one protocol data unit having at least one field containing the value and the step of retrieving a value from the network frame further includes;

retrieving protocol knowledge of the data structure of the at least one protocol data unit enabling the extraction of the at least one field; and

extracting a value from the at least one field of the at least one protocol data unit.

15. The method according to any of claims 13 and 14 wherein the step of performing an action includes:

receiving at least one protocol data unit name and at least one value corresponding to at least one keyword associated with the at least one protocol data unit name;

retrieving protocol knowledge of the data structure of a network frame enabling the building of the at least one keyword into a network frame;

associating the at least one value with the at least one keyword;

placing the at least one protocol value into a memory device in a network frame data structure; and

transmitting the network frame to the network.

16. A computer readable memory system encoded with a protocol finite state machine, the protocol finite state machine including:

a set of state keywords defining each state of the protocol that the protocol may enter, the set of state keywords representing a number of behavior states that a particular one of the protocol states may express, the particular one of the protocol states being associated with a particular one of the state keywords; and

a set of event keywords defining each state keyword representative of inputs that the protocol state may receive while being associated with the particular one of the state keywords, the protocol state operable to change its association to another one of the state keywords in response to the inputs received.

FOI 2010-0404850